219.40057X00
P11631

UNITED STATES PATENT APPLICATION FOR:

FAULT RESILIENT BOOTING FOR MULTIPROCESSOR SYSTEM USING APPLIANCE SERVER MANAGEMENT

Inventor:

Son H. LAM

Prepared by:
Antonelli, Terry, Stout & Kraus, LLP
Suite 1800
1300 North Seventeenth St.
Arlington, VA 22209

Phone: 703-312-6600
Fax: 703-312-6666

FAULT RESILIENT BOOTING FOR MULTIPROCESSOR SYSTEM USING APPLIANCE SERVER MANAGEMENT

FIELD

The present invention is directed to a system for booting a multiprocessor computer system. More particularly, the present invention is directed to a system for booting a multiprocessor computer system using an appliance server management driver.

BACKGROUND

The use of fault resilient booting is known in the art and for example is described in U.S. Patent 5,790,850. As described therein and as shown in Fig. 1, a multiprocessor system includes a number of processors 10-13 each of which include a local advance programmable interrupt controller (APIC) 14-17. The local APIC units are connected through an APIC 19 bus. An input/output APIC unit 28 is also connected to this bus. A processor bus 20 connects the processors and the memory.

In this system, when power is initially applied to the processors one of the processors is designated the bootstrap processor. One of the processors can be designated in the hardware for this function. The other processors are classified as application processors. Each of the processors undergoes a built in self test when power is initially applied. If the processor is faulty for any reason, it stores a status flag to indicate this. If the bootstrap processor is faulty, it is necessary to designate one of the application processors to handle the bootstrap function instead. U.S.

Patent 5,790,850 shows one method for doing this where application processors that have been tested to be good are successively examined. If all tests are passed, that application processor is designated as the bootstrap processor and that function is removed from the original bootstrap processor.

In systems of this type, the fault resilient booting is implemented in servers using the basic input output system (BIOS), the baseboard management controller (BMC) and other hardware to follow this procedure when the bootstrap processor fails. Most of this function is implemented in the baseboard management controller chip. However, the inclusion of this chip adds to the cost of the system. While this is not a problem for more expensive systems, in low cost servers, it is desirable to reduce the cost of the system.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and a better understanding of the present invention will become apparent from the following detailed description of example embodiments and the claims when read in connection with the accompanying drawings, all forming a part of the disclosure of this invention. While the foregoing and following written and illustrated disclosure focuses on disclosing example embodiments of the invention, it should be clearly understood that the same is by way of illustration and example only and that the invention is not limited thereto. The spirit and scope of the present invention are limited only by the terms of the appended claims.

The following represents brief descriptions of the drawings, wherein:

Figure 1 is an example background arrangement useful in gaining a more thorough understanding and appreciation of the present invention;

Figure 2 is an example advantageous embodiment of the present invention.

DETAILED DESCRIPTION

Before beginning a detailed description of the subject invention, mention of the following is in order. When appropriate, like reference numerals and characters may be used to designate identical, corresponding or similar components in differing figure drawings. Further, in the detailed description to follow, example sizes/models/values/ranges may be given, although the present invention is not limited to the same. With regard to description of any timing signals, the terms assertion and negation may be used in an intended generic sense. More particularly, such terms are used to avoid confusion when working with a mixture of "active-low" and "active-high" signals, and to represent the fact that the invention is not limited to the illustrated/described signals, but could be implemented with a total/partial reversal of any of the "active-low" and "active-high" signals by a simple change in logic. More specifically, the terms "assert" or "assertion" indicate that a signal is active independent of whether that level is represented by a high or low voltage, while the terms "negate" or "negation" indicate that a signal is inactive. As a final note, well known power/ground connections to ICs and other components may not be shown within the FIGS. for simplicity of illustration and discussion, and so as not to obscure the invention. Further, arrangements may be shown in block diagram form in order to avoid obscuring the invention, and also in view of the fact that specifics with respect to implementation of such block diagram arrangements are

highly dependent upon the platform within which the present invention is to be implemented, i.e., such specifics should be well within purview of one skilled in the art. Where specific details (e.g., circuits, flowcharts) are set forth in order to describe example embodiments of the invention, it should be apparent to one skilled in the art that the invention can be practiced without, or with variation of, these specific details. Finally, it should be apparent that differing combinations of hard-wired circuitry and software instructions can be used to implement embodiments of the present invention, i.e., the present invention is not limited to any specific combination of hardware and software.

As shown in U.S. Patent 5,790,850, a fault resilient booting scheme involving a plurality of processors can be accomplished by successively examining each application processor if the original bootstrap processor fails. Figure 1 shows the overall arrangement of the processors and other major parts of such a system. Previously, the controls for the fault resilient booting process were stored in a baseboard management controller chip. The present invention avoids the necessity of having this chip by relying on an appliance server management arrangement so that the system can be provided at a lower cost.

The fault resilient booting processor includes at least three levels or parts that are controlled by timers at different stages of the basic input output system (BIOS) before the system is handed over to the operating system. The table below describes the different fault resilient booting levels and describes the time line by which they are executed.

Reset de-asserted (BIST execution)	**FRB-3 started**
BIOS code execute	
Check built in self test result	**FRB-1**
Power on self test start	**FRB-2 started**
	FRB-3 reset
Power on self test exit	**FRB-2 reset**

This table describes the three levels of the fault resilient booting process which must be encountered before the system is handed over to the operating system for normal operations. The FRB-3 level refers to the portion of the process where a timer is started upon the power-up of the system or upon a hard reset. This timer must be stopped by BIOS. This requires the bootstrap processor to actually run BIOS code. If the timer is stopped this indicates that the bootstrap processor can actually run code and accordingly is not dead at this time. If the timer is not stopped, the bootstrap processor is disabled, the system is reset and another processor is assigned to become the bootstrap processor. When a new bootstrap processor is assigned, the APIC identification is changed so that the second processor is identified as the bootstrap processor. The BIOS running in the bootstrap processor is responsible for stopping the FRB-3 timer during a power on self test. This is accomplished by resetting the watchdog timer which is producing the timing signals.

The next level of the fault resilient booting, FRB-2, involves the use of the watchdog timer to backup the operation of the baseboard management controller during the power-on self test. BIOS sets a bit in the baseboard management

controller to indicate that BIOS is in the FRB-2 phase. This bit is set after it is determined which processor is the bootstrap processor. BIOS then sets the FRB-2 bit, loads the watchdog timer with a new time-out interval and disables FRB-3. Using this process, there is no gap in the watchdog timer coverage between FRB-3 and FRB-2. If the FRB-2 phase is successful, BIOS disables the FRB-2 time-out prior to exiting the power on self test. The baseboard management controller provides commands for this purpose. This is generally done prior to initiating the option ROM scan.

If the timer expires during the FRB-2 function, the baseboard management controller generates a FRB-2 time-out message and hard resets the system. BIOS then determines that the previous boot attempt failed FRB-2 and examines the FRB-2 time out flag. BIOS then issues a disable processor command in order to disable the CPU that had failed the FRB-2 test.

The FRB-1 level is implemented by BIOS. If the bootstrap processor has failed, BIOS records the events so they can be logged later and disables the processor by sending a command to the baseboard management controller.

As can be seen in this description, the baseboard management controller is used to control this testing procedure using BIOS and the processors of the system. However, the inclusion of this chip causes additional cost for the system. It is desirable to eliminate this chip for less expensive systems. This can be accomplished with an appliance server management system.

The appliance server management system is an architecture utilizing arrangement of hardware, drivers, providers and software. This type of system can

be implemented in a number of computers and includes a number of desirable features, including the ability to manage the system either locally or over the internet. It also allows the automatic alerting of an administrator when problems arise with the system. In regard to the present situation, it is possible to implement a fault resilient booting system using this management system. In doing so, it is possible to eliminate the baseboard management controller chip and instead use existing parts of the ASM system instead.

The following description helps to describe the difference in implementation using an ASM system rather than a BMC system. In the FRB-3 level the timer is in the BMC and is programmable or will assume a default time of ten seconds. This timer starts upon a power up or a hard reset. This timer must be stopped by BIOS by resetting the timer. If the timer expires, a signal is sent to the failed processor to indicate that it cannot act as the bootstrap processor and an internal message is generated indicating the failure. In this same level, the ASM system uses an on-board watchdog timer which is set to six seconds since BIOS operations are normally completed in less than five seconds. This timer is automatically started after the system resets. If the timer expires, it will set the CPU STOP Latch which sends a signal to disable the bootstrap processor.

In regard to the FRB-2 level, the BMC system uses the timer in the BMC while the ASM system uses a timer in the system I/O chip (SIO). In BMC, if the timer expires an event message is generated and the system is hard reset. BIOS then determines that the attempt has failed and sends a command to BMC to disable the failed processor. In ASM, when the timer expires, BIOS disables the bootstrap

processor by setting a CPU STOP Latch by way of a general purpose I/O bit from the S1O chip.

In the FRB-1 level, in BIOS checks a processor built in self test (BIST) result. If the bootstrap processor fails, BIOS will assign this function to another processor. In ASM, if a built-in self test failure occurs, BIOS takes its own steps to record the event so that it can be logged later. BIOS disables the processor by setting the CPU STOP Latch by way of a general purpose I/O bit from the SIO chip. The latch can only be reset by another signal from the SIO chip. If BIOS is unable to set the CPU STOP Latch then the FRB-3 timeout is allowed to occur.

Figure 2 shows part of the hardware 50 utilized in the ASP system to control the fault resilient booting process. The system I/O chip (SIO) 52 provides many of the enabling signals for this process. Each of the outputs of this chip are labeled as general purpose I/O (GPIO) signals. This chip is programmed to follow the process and to provide the control signals based on the implementation described above.

The watchdog timer (WDT) 54 provides an output at six seconds so that this timer may be used for the FRB-3 test. When this signal is generated, the timer is considered to have expired and the signal is applied to OR gate 56 and then passed to the set input of CPU STOP Latch 58. This latch is set by the occurrence of this signal and the signal is then sent to disable CPU 60 which is the initial bootstrap processor.

The WDT produces an output signal after six seconds, as discussed above. The start of this six second period occurs due to the arrival of GPIO4 from the SIO or

system reset which is applied to OR gate 53 which resets the timer. The SIO generates the signal due to the power being turned on or to a reset signal.

When the six second signal is sent to OR gate 56, it is also necessary to reset the system so that a second processor can be considered for the bootstrap operation. Accordingly, the six second signal also is branched off to OR gate 55 to cause a system reset. Other reset signals can be applied to OR gate 55 also. A second input (GPIO 4) to OR gate 53 can also reset the timer. The CPU 62 is an application processor which can be disabled through the S10's GP10 3.

The CPU STOP Latch 58, once set, can only be re-set by the receipt of a GPIO2 signal from the SIO at the reset input. Thus, this latch is not merely reset from a reset signal, but must be specifically opened by the SIO in view of the system condition.

In the FRB-2 level testing, the timer in the SIO chip is utilized to determine if the FRB-2 function has failed by the end of the timeout period. If the FRB-2 level test is failed, the signal GPIO 1 is generated and applied as an input to OR gate 56. It is then passed to the set position of the CPU STOP Latch which then turns off CPU 1 to prevent it from operating as the bootstrap processor.

Likewise, if the FRB-1 level test indicates a failure, the SIO generates GPIO1 signal which is passed to the CPU STOP Latch 58 to disable CPU 60.

Thus, it can be seen how this arrangement of hardware can produce the fault resilient booting process according to the ASM system using the test as described above. In so doing, the baseboard management controller chip is unnecessary and

instead the ASM architecture is able to perform these tests under the control of ;the SIO.

In concluding, reference in the specification to "one embodiment", "an embodiment", "example embodiment", etc., means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of such phrases in various places in the specification are not necessarily all referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with any embodiment, it is submitted that it is within the purview of one skilled in the art to effect such feature, structure, or characteristic in connection with other ones of the embodiments. Furthermore, for ease of understanding, certain method procedures may have been delineated as separate procedures; however, these separately delineated procedures should not be construed as necessarily order dependent in their performance, i.e., some procedures may be able to be performed in an alternative ordering, simultaneously, etc.

Further, the present invention may be practiced as a software invention, implemented in the form of a machine-readable medium having stored thereon at least one sequence of instructions that, when executed, causes a machine to effect the invention. With respect to the term "machine", such term should be construed broadly as encompassing all types of machines, e.g., a non-exhaustive listing including: computing machines, non-computing machines, communication machines, etc. Similarly, which respect to the term "machine-readable medium", such term should be construed as encompassing a broad spectrum of mediums, e.g., a non-

exhaustive listing including: magnetic medium (floppy disks, hard disks, magnetic tape, etc.), optical medium (CD-ROMs, DVD-ROMs, etc), etc..

This concludes the description of the example embodiments. Although the present invention has been described with reference to a number of illustrative embodiments thereof, it should be understood that numerous other modifications and embodiments can be devised by those skilled in the art that will fall within the spirit and scope of the principles of this invention. More particularly, reasonable variations and modifications are possible in the component parts and/or arrangements of the subject combination arrangement within the scope of the foregoing disclosure, the drawings and the appended claims without departing from the spirit of the invention. In addition to variations and modifications in the component parts and/or arrangements, alternative uses will also be apparent to those skilled in the art.